AMENDMENTS TO THE SPECIFICATION

Please delete the paragraphs beginning at page 18, line 18 through page 20, line 9, and replace as follows:

If the execution system determines that the current user action(s) cannot be aligned with the procedure (negative determination in step 310), the system will continue to obtain recorded user actions (step 309) and attempt alignment (step 311). As the process of real-time alignment of user actions with the procedure continuously fails, the execution system will continue to record the user actions until the user terminates the sub-procedure execution (affirmative determination in step 314).

If the execution system determines that the current user action(s) can be aligned with the procedure (affirmative determination in step 310), the execution system queries the user as to whether the system should take control of the execution (step 311). If the user does not relinquish control of the procedure execution to the system (negative result in step 311), the system will continue to record and process the user actions. On the other hand, if the user allows the execution system to regain control of the procedure execution (affirmative result in step 311), the system will stop recording the user actions (312).

The recorded actions will then be processed to augment the learned procedure (step 313) so that the same situation encountered during future invocations of the procedure results in a normal procedure execution. In one embodiment of the invention, step 313 is performed by sending the recorded user actions to the procedure repository (205), whereby the recorded user actions are processed by the server (206) to update the executable procedure stored in database (209) of the procedure server (208), as shown in FIG. 2.

In one preferred embodiment according to the present invention, when an executable procedure is performed in a GUI (graphical user interface)-based system, such as the Microsoft Windows® operating systems or a Java® virtual machine, the executed actions (step 306, FIG. 3) are performed by simulating user actions such as, mouse and keyboard clicks. This can be done by invoking the appropriate API of the GUI, for example, APIs described in Microsoft Development Network ("MSDN") on-line resources.

Advantageously, the method of FIG. 3 enables executable procedures to be learned from examples incrementally. Initially, executable procedures that are learned from a handful of examples (execution traces) can be distributed and widely executed. Indeed, paths through a procedure that apply to large number of users can be quickly learned, and then disseminated to a large portion of the user base, which users can readily benefit from the learned procedure. Then, when new paths (new sub-procedure) of the procedure are subsequently encountered, systems and methods according to the present invention (e.g., the method of FIG. 3) can be implemented to enable a user to manually perform the new sub-procedure. The manually entered sub-procedure results in execution traces that can be used for learning/augmenting/updating an executable procedure, and incorporated in the executable procedure.

The execution of a new sub-procedure may include calling a support center, but the length of such call would generally be shortened/reduced (as compared to conventional methods), because the execution system could regain control of the procedure execution after the sub-procedure terminates, and then automatically execute known (learned) actions. Moreover, an updated, executable procedure can be reused by other users. Therefore, the number and length of calls to the service center to solve the same problem is substantially reduced using the systems and methods described herein.